Actes de l'atelier HIA 2020: Humains et IA, travailler en intelligence

David Gross-Amblard

Actes de l'atelier HIA 2020: Humains et IA, travailler en intelligence

28 janvier 2020, Bruxelles
Atelier de la conférence EGC 2020

Site web de l'atelier :
http://headwork.gforge.inria.fr/HIA2020

Actes en ligne : https://hal.inria.fr/

Table des matières

1 Comités

Président du comité de Programme
David Gross-Amblard, Univ Rennes, IRISA

Comité d'organisation
David Gross-Amblard, Univ Rennes, IRISA

Comité de Programme
Tristan Allard, Univ Rennes, IRISA
Sihem Amer-Yahia, CNRS, LIG
Sandro Bimonte, TSCF, IRSTEA
Pierre Bourhis, CNRS, Cristal
Baptiste Caramiaux, CNRS, LRI
François Charoy, Inria, Loria
Karen Fört, Université Paris Sorbonne, STIH
David Gross-Amblard, Univ Rennes, IRISA
Stéphane Huot, Inria, Cristal
Odalric-Ambrym Maillard, Inria, Cristal
Arnaud Martin, Univ Rennes, IRISA
Paola Tubaro, CNRS, LRI
Karine Zeitouni, UVSQ, David

Edition des actes
David Gross-Amblard, Univ Rennes, IRISA

Explicabilité : vers des dispositifs numériques interagissant en intelligence avec l'utilisateur

Pierre-Antoine Champin*, Béatrice Fuchs*
Nathalie Guin*, Alain Mille *,**

*Université Lyon, Université Lyon1, CNRS, LIRIS, F-69622
**Coexiscience, F-69100

Résumé. Cet article de positionnement postule que pour qu'un utilisateur et un dispositif technique numérique soient *en intelligence*, il est nécessaire de penser l'explicabilité. Nous proposons une définition de l'explicabilité fondée sur l'objectif d'appropriation de tout dispositif technique numérique par l'utilisateur. L'explicabilité nécessite l'explicite, mais aussi les processus d'exploration des régulations conditionnant le comportement des dispositifs techniques numériques. Après avoir passé en revue ce qui est explicité ou non par les dispositifs concernés (en particulier lorsqu'il s'agit de dispositifs dits intelligents), nous proposons de soutenir les processus d'explication par l'usage des traces d'interaction comme matériau à modéliser avec l'utilisateur jusqu'à pouvoir *aligner* ou *mettre en congruence* sa compréhension avec la réalité du fonctionnement du dispositif, dans le contexte de son activité. Nous soutenons que cette approche est très prometteuse pour améliorer l'explicabilité, y compris lorsque l'explicité n'est pas accessible ou peut-être même n'existe pas à l'avance.

1 Introduction

L'équipe TWEAK a comme objectif de recherche l'étude des conditions permettant aux dispositifs techniques numériques d'interagir *en intelligence* avec l'utilisateur. Au delà du développement de dispositifs *intelligents* il s'agit aussi que ces dispositifs facilitent leur appropriation par les utilisateurs. Cette appropriation suppose un apprentissage encapacitant de l'utilisateur pour qu'il puisse *agir* et adapter le dispositif à son propre contexte, à ses propres connaissances, à ses propres objectifs. La nécessité de cette appropriation poursuit des objectifs d'efficacité dans les usages (Liquète et al., 2012) mais aussi de capacité éthique (Mille, 2019). Pour qu'un processus d'appropriation puisse advenir pendant l'interaction avec un dispositif technique numérique, il est nécessaire que les régulations encapsulées dans ce dispositif, régissant la manière dont il a été préparé pour réagir aux différentes situations rencontrées, soient *explicables* lors de leur mise en œuvre par l'utilisateur.

Nous proposons de considérer les traces d'interaction comme le matériau de départ pour les processus d'explication. En effet, une trace d'interaction contient des éléments combinés issus des fonctions du dispositif et des actions de l'utilisateur. Formalisées, ces traces permettent de mener des calculs pour retrouver les schèmes de régulation dans les motifs reconnus en

interaction avec l'utilisateur. C'est l'utilisateur qui a l'initiative de guider la découverte de connaissance en cherchant à reformuler les interactions à un niveau d'abstraction rejoignant la manière dont il décrit sa propre activité. Depuis plus de 10 ans maintenant, l'étude du potentiel des *traces modélisées* est menée dans différents domaines, avec des fonctions d'assistance à l'appropriation des dispositifs et de leurs régulations (Champin et al., 2013).

Pour poser le problème de l'explicabilité, nous rappellerons comment elle se décline dans le domaine de l'informatique à l'ère de ce que l'on appelle couramment les *intelligences artificielles*, que nous situerons rapidement. Le problème posé, nous exposerons l'approche que nous adoptons dans nos recherches visant à mettre en intelligence l'utilisateur et son environnement informatique par des dispositifs menant à établir une *congruence* d'interprétations. En d'autres termes, nous étudions comment ajuster la compréhension de l'utilisateur et la sémantique des régulations intégrées dans le dispositif technique numérique. Après avoir présenté les modèles et les outils que nous développons pour les traces modélisées, nous illustrerons leur mobilisation sur plusieurs cas d'usage où les fonctions *d'explicabilité*, telles que nous les avons définies, ont été conçues, mises en œuvre et exploitées : assistance à la construction interactive d'un processus d'analyse de traces ; explicitation des processus d'analyse de traces pour leur appropriation en réutilisation ; exploration interactive de traces avec assistance à leur modélisation ; appropriation des fonctions de traçage et négociation de la construction d'indicateurs d'apprentissage. Nous conclurons cet article de positionnement en montrant le potentiel de cette approche et en traçant les perspectives ouvertes aussi bien théoriques que techniques que nous souhaitons discuter dans la communauté de *l'explicabilité* telle qu'elle commence à se former.

2 Intelligence, informatique, intelligence artificielle

C'est avec la conférence de Dartmouth en 1956, et la création du terme « Intelligence Artificielle », que la notion d'intelligence est appliquée à l'informatique. Il s'agissait alors de caractériser des dispositifs de calcul qui puissent émuler voire dépasser les fonctions cognitives, par exemple [1] : les fonctions exécutives, les fonctions visuo-spatiales, les gnosies, le langage, la mémoire, les praxies, et ceci en cherchant à défier la vitesse de traitement de l'information dans le cerveau.

L'examen de ces fonctions montre qu'elles ne sont pas indépendantes. Pourtant, elles ont été et sont encore souvent étudiées séparément, comme si elles n'avaient pas d'interactions les unes avec les autres dans leur fonctionnement. Au départ, la question principale était de découvrir ce que le cerveau (essentiellement) effectue comme traitement pour réaliser ces fonctions cognitives. Les fonctions exécutives ont été étudiées comme des formes de raisonnement formels ayant pour objet de déterminer la meilleure solution à un problème. L'hypothèse était faite que les problèmes étaient formalisables et que leur résolution était sans doute calculable (Newell et al., 1959). Les interactions fournissent des flux d'information entrant et sortant : des données à traiter et produisant d'autres données. Les fonctions cognitives autres que le raisonnement sont à l'évidence indissociables des possibilités d'interaction mêmes. Pour les étudier, elles sont souvent sorties de leur contexte corporel vivant en les considérant dans des contextes de corps artificiels (robots). Les applications de ces recherches consistent à pouvoir déléguer

1. Liste issue de `https://aqnp.ca/la-neuropsychologie/les-fonctions-cognitives/`

telle ou telle fonction cognitive humaine à un dispositif de calcul qui pourrait assurer cette fonction au moins aussi bien que l'humain. Une application, un dispositif technique numérique réalisé selon ces principes, sera nommé *intelligence artificielle* dès l'instant où une fonction cognitive humaine lui est déléguée. Les applications de recommandation, les systèmes de décision, les systèmes de diagnostic, de pronostic, de planification, d'organisation, de traduction,... sont légion. L'essentiel de l'informatique en interaction avec les humains pourrait alors relever de cette définition.

La question de la régulation [2] (Simondon, 1989) embarquée dans ces fonctions se pose : quelle est la sémantique de régulation exprimée *via* les règles codées ? Deux possibilités : les règles sont formellement décrites, en tant que données, pour être traitées par des algorithmes d'inférence ; les règles sont codées directement dans la fonction. Dans le premier cas, leur explicitation est possible par construction, à l'image du système expert pionnier Mycin (Buchanan et Shortlife, 1984). Dans le second cas, les règles codées peuvent être représentées de manière explicite dans un langage de programmation, ou cristallisées dans la structure même du dispositif, par exemple des réseaux neuronaux. Les règles sont associées aux informations, comme mode d'emploi. Cette association constitue une *connaissance* : le dispositif déclenchant la fonction sait quoi faire d'une information disponible. La mise en œuvre de ces connaissances réalise une sorte d'intelligence artificielle. L'apprentissage et la gestion de ces règles sont étudiées en tant qu'ingénierie des connaissances (Newell, 1982). Dans le cas des règles explicites exploitées par un moteur de raisonnement, l'ingénierie des connaissances propose des processus de collecte, représentation, gestion et exploitation sous des formes explicites. Le Web Sémantique s'inscrit dans cette catégorie. Dans le cas du codage par un programmeur directement dans l'algorithme, seul le code est explicite. Dans le cas de règles apprises par des dispositifs d'apprentissage automatique, on distinguera les approches symboliques, des approches non symboliques (Cornuéjols et al., 2018). Dans les deux cas, il est toujours très difficile de connaître la justification des règles, car le processus de collecte des données d'apprentissage n'est pas embarqué avec son résultat.

L'*explicabilité* des dispositifs techniques numériques est devenu un enjeu de société, à tel point que lorsque leur fonctionnement est régi par des règles issues des algorithmes d'apprentissage profond, elle donne lieu à des débats animés. [3] Dans cet article, nous considérons qu'un dispositif est explicable *lorsqu'il réunit les conditions pour permettre à l'utilisateur de s'approprier la sémantique des régulations à l'œuvre dans ce dispositif.* Cette définition impose au dispositif de pouvoir exploiter les descriptions explicites des règles disponibles dans l'environnement numérique, mais aussi de permettre à l'utilisateur d'explorer, à son initiative, le dispositif en fonctionnement. L'explicabilité ne dépend donc pas uniquement de l'explicitation mais aussi de la capacité à la projeter dans l'expérience utilisateur. La section suivante s'attache à définir plus précisément cette notion de « en intelligence ».

2. Tout objet technique est conçu avec des règles d'usage que le *technicien* doit comprendre pour le maîtriser. Si la règle d'usage d'un marteau peut se deviner en le manipulant, la règle d'usage d'un dispositif numérique se révèle bien plus difficilement, d'autant que le résultat de l'usage n'est pas observable directement

3. L'alerte dans un journal de vulgarisation scientifique `https://www.sciencesetavenir.fr/high-tech/intelligence-artificielle/intelligence-artificielle-l-explicabilite-talon-d-achille-du-deep-learning_124903` ; la théorisation de la question en relation avec la capacité à respecter la loi (RGPD par exemple) : `https://perso.math.univ-toulouse.fr/mllaw/home/statisticien/explicabilite-des-decisions-algorithmiques/`

3 Interagir en intelligence : congruence des interprétations

Nous avons vu comment des intelligences artificielles (IA dans la suite) pouvaient se mettre en place avec des capacités d'explicitation variables selon la technique utilisée. Dans le scénario d'usage d'une IA, l'explicabilité est dépendante de la conception de cette IA : soit la conception a prévu des fonctions d'explication exploitant les capacités intrinsèques d'explicitation de la technique utilisée, pour fournir les informations nécessaires (informations mobilisées, règles formelles déclenchées,...). L'utilisateur peut ainsi être informé *via* la documentation produite avec une possible mise en contexte – par exemple en fournissant une visualisation du graphe de relations sémantiques mobilisé pour produire un résultat (Hasan, 2014) ; soit ce n'est pas prévu, et l'utilisateur ne peut que tenter de deviner les régulations à l'œuvre en testant l'application avec des informations différentes, et par induction les imaginer. Quand il n'y a pas de substrat explicite (cas des réseaux neuronaux appris), des mécanismes dédiés peuvent être proposés pour re-construire du sens a posteriori à partir des régularités découvertes par la machine (Olah et al., 2018).

Mais quels que soient les efforts de conception pour rendre les dispositifs ainsi explicables, il demeure un problème de congruence [4] entre les constructions élaborées par le dispositif et celles compréhensibles par l'utilisateur. Cette congruence pose un problème dans les deux sens : difficulté pour l'utilisateur de comprendre une explication produite par le dispositif, difficulté pour le système d'adopter le registre [5] habituel d'expression de l'utilisateur (par exemple une expression en langue naturelle) pour son propre registre formel (par exemple une requête en langage informatique).

De nécessaires reformulations s'imposent pour passer d'un registre à l'autre, jusqu'à obtenir un effet de couplage entre le dispositif et l'utilisateur. La notion de *couplage* renvoie à l'idée qu'après avoir compris comment fonctionnait un système, il n'est plus besoin de réfléchir pour l'utiliser : on le fait *sans y penser*.

Comment alors construire des dispositifs ayant cette capacité d'explicabilité ? L'explicabilité étant une condition nécessaire (bien que non suffisante) pour définir des dispositifs techniques que l'utilisateur puisse s'approprier, nous reprenons et précisons ici les propriétés à donner à un dispositif pour lui conférer cette capacité d'explicabilité : 1) Caractère explicite des règles à l'œuvre dans les fonctions activées lors de l'utilisation d'un dispositif. Une fonction est explicite lors de son activation si elle dispose d'un mécanisme de description symbolique des règles à l'œuvre [6] ; 2) Propriété d'explicabilité des régulations à l'œuvre dans le contexte d'usage d'un utilisateur. Une fonction est explicable si l'utilisateur peut interagir avec un assistant d'explication permettant de mettre en relation l'expérience utilisateur liée au contexte d'usage et les informations explicitées disponibles.

Pour rendre *explicable* un dispositif technique numérique, nous proposons d'utiliser les traces d'utilisation comme information complémentaire pour co-construire les explications en interagissant avec l'utilisateur. Les traces d'utilisation sont articulables avec les explicitations

4. Dans ce texte, nous utilisons le terme congruence pour indiquer la possibilité d'alignement entre les interprétations (sémantique) faites d'une expression symbolique partagée (sémiotique).

5. Le registre d'expression est constitué de l'ensemble des éléments disponibles pour exprimer une information sous une forme symbolique explicite correspondant à la sémantique recherchée.

6. Cette propriété est très utile, mais pas garantie car dépendant des efforts de conception pour développer une facette explicitation de fonction. C'est objectivement difficile avec les outils de développement classique (applications compilées), c'est possible lorsque l'application exploite un moteur d'inférence associé aux connaissances représentées explicitement, c'est aujourd'hui presque impossible pour les fonctions représentées par un réseau neuronal appris.

disponibles dans l'environnement. Le rapprochement entre le point de vue utilisateur et le point de vue concepteur relève de la mise en congruence d'interprétations sur des formes sémiotiques. C'est la *congruence des interprétations* qui autorise le travail *en intelligence.*

4 Les traces *modélisées* comme conteneur de connaissances

L'équipe TWEAK du laboratoire LIRIS a proposé un méta-modèle pour la construction de systèmes à base de traces, des systèmes exploitant les connaissances présentes dans les traces d'interaction entre l'utilisateur et le système (Champin et al., 2013). La notion centrale de ce méta-modèle est celle de m-trace (*modeled trace*), définie comme une liste d'éléments observés appelés obsels (*observed elements*). Chaque obsel est décrit par un type, un ensemble d'attributs, et deux estampilles temporelles début et fin, délimitant l'intervalle durant lequel cet obsel a pu être observé [7]. Chaque m-trace est associée à un modèle de traces, qui spécifie les types d'obsels que la trace peut contenir, ainsi que les attributs de chaque type d'obsels. Ainsi, le modèle de traces permet d'expliciter la structure et la sémantique sous-jacente d'une m-trace. Cette connaissance est capitalisable, puisque plusieurs m-traces décrivant des activités similaires peuvent faire référence au même modèle. On peut par exemple imaginer un modèle de traces décrivant les traces d'interaction issues d'une plateforme d'enseignement à distance qui propose des cours et des exercices. Il pourrait ainsi y avoir trois types d'obsels : la *consultation d'un cours*, la *réponse à un exercice*, la *demande d'une aide* pendant un exercice. Ces trois types d'obsels partageront un attribut *identifiant l'apprenant*. Le type *consultation d'un cours* pourra avoir un attribut *identifiant le cours*. Le type *réponse à un exercice* aura un attribut indiquant l'*identifiant de l'exercice*, un attribut précisant le *type d'exercice* (QCM, QROC, appariement, etc.), un attribut indiquant à quel *cours* il se rapporte, un attribut décrivant son *niveau de difficulté*, et un attribut indiquant le *feedback* fourni par le système (par exemple réponse correcte ou incorrecte). Les m-traces sont stockées et traitées par un Système de Gestion de Traces Modélisées (SGTM). Un SGTM contient deux types de m-traces : les m-traces premières qui contiennent des obsels collectés directement depuis les applications ; et les m-traces transformées, dont les obsels sont calculés à partir du contenu d'une (ou plusieurs) m-trace(s) source(s), qui peuvent être indifféremment premières ou transformées. Les transformations sont notamment utilisées pour « élever » la description de l'activité d'un modèle bas niveau (focalisé sur les interactions *atomiques* [8]) vers un modèle de plus haut niveau (décrivant des actions ou des tâches plus abstraites).

kTBS [9] (kernel for Trace-Based Systems) est une implantation de référence open-source de SGTM. Il utilise le modèle de données RDF qui offre la flexibilité nécessaire pour représenter les m-traces selon divers modèles de traces. kTBS fournit un ensemble d'opérateurs de transformation, depuis de simples filtres jusqu'à des ré-écritures complexes spécifiées en SPARQL. Il permet également la définition d'opérateurs personnalisés. Techniquement, kTBS offre une API REST qui permet de créer, consulter et modifier les m-traces et leurs modèles.

7. Il est toujours possible que début et fin aient la même valeur, pour représenter des événements instantanés.
8. Une interaction est atomique lorsqu'elle n'est pas décrite en sous-interactions ; elle peut toutefois concerner une interaction décrite à un niveau très abstrait.
9. `http://tbs-platform.org/tbs/doku.php/tools:ktbs`

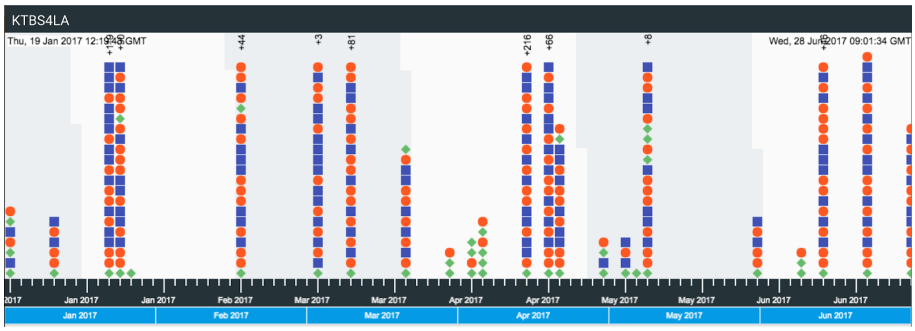


FIG. 1 – *Visualisation sous forme de timeline de l'activité des apprenants. Chaque obsel est représenté par un symbole dépendant des critères définis par l'utilisateur analyste (en fonction de son type et de ses valeurs d'attributs).*

5 L'explicabilité par les traces : illustrations

Permettre à un analyste de construire de manière interactive un processus d'analyse de traces. La plateforme kTBS présentée en section 4 est destinée à l'analyse de traces d'interaction. Dans le contexte des outils numériques pour l'éducation ou la formation, l'analyse des interactions entre les apprenants et l'environnement d'apprentissage (Learning Analytics) est nécessaire pour suivre la progression des apprenants, leur apporter un soutien, et mieux comprendre les mécanismes d'apprentissage. Les utilisateurs ayant besoin d'analyser les traces ne sont en général pas des spécialistes des technologies du web sémantique et ne peuvent pas exprimer des requêtes en SPARQL. Nous avons donc conçu la plateforme d'analyse de traces kTBS4LA [10] (kTBS for Learning Analytics), qui permet à un utilisateur de définir un modèle des traces qu'il souhaite analyser, de visualiser ces traces pour les explorer, d'effectuer des transformations de traces et de calculer des indicateurs sur l'activité des apprenants. Les spécificités de cet outil sont l'importance donnée à la temporalité des traces, une représentation explicite de la sémantique des interactions tracées, et la possibilité pour l'utilisateur de manipuler directement les données qu'il analyse. Pour analyser les traces issues d'un environnement d'apprentissage, l'analyste commence en effet par décrire ces traces en définissant leur modèle ; il donne ainsi du sens aux interactions tracées. Ce modèle de traces est utilisé pour proposer à l'analyste une visualisation des traces dans laquelle les obsels de différents types sont représentés différemment (cf. Figure 1). L'utilisateur analyste peut définir des règles décrivant quelle forme et quelle couleur utiliser pour représenter un obsel vérifiant certaines contraintes portant sur son type ou ses attributs. Par exemple, il pourra choisir de filtrer les données en faisant disparaître les obsels de type *consultation d'un cours*, de faire apparaître comme un losange vert les obsels de type *demande d'aide*, comme un carré bleu les obsels de type *réponse à un exercice* dont le *niveau de difficulté* est supérieur à un seuil et comme des ronds rouges ceux dont le *niveau de difficulté* est inférieur à ce seuil. Cet exemple de visualisation peut servir à l'utilisateur pour voir si les apprenants demandent davantage d'aide dans les exercices difficiles que dans les exercices plus faciles. L'ensemble des règles de visualisation ainsi définies par l'utilisateur sur une trace donnée constitue une feuille de style qui peut être

10. http ://tbs-platform.org/tbs/doku.php/tools :ktbs4la

enregistrée pour être ré-utilisée sur une autre trace ayant le même modèle. Différentes feuilles de style peuvent être utilisées pour visualiser les traces, pour répondre à différents besoins d'analyse. Une feuille de style peut également servir à créer un opérateur de transformation qui permettra de créer de nouvelles traces représentant un nouveau point de vue sur l'activité des apprenants. Cet opérateur de transformation pourra être appliqué à un ensemble de traces ayant le même modèle, même s'il a été défini initialement sur une trace donnée. Si l'on reprend l'exemple ci-dessus, les traces transformées n'auront plus de type d'obsels *consultation d'un cours* ni *réponse à un exercice*, et auront deux nouveaux types d'obsels : *réponse à un exercice facile* et *réponse à un exercice difficile*. La plupart du temps, les analyses sur les données d'apprentissage sont effectuées par des informaticiens et restituées aux équipes pédagogiques ou aux chercheurs sous forme de tableaux de bord qui ne correspondent pas toujours aux besoins des utilisateurs, ou à l'évolution de ces besoins (lorsque que les utilisateurs savent exprimer leurs besoins), ce qui nécessite de nouvelles analyses et la conception de nouveaux tableaux de bord. Avec kTBS4LA, l'analyste manipule directement les traces et peut construire de manière dynamique et interactive le processus d'analyse répondant à ses besoins. En permettant à l'utilisateur de donner une sémantique explicite aux traces, en lui permettant de construire lui-même les visualisations et donc les interprétations des données dont il a besoin, le mécanisme d'analyse est explicite et ses résultats sont explicables. On peut dans ce cas parler d'explicabilité *by design*.

Documenter des processus d'analyse de traces pour faciliter leur réutilisation. Dans la communauté des Learning Analytics, la question de la capitalisation des analyses est posée. D'un point de vue informatique, ces analyses sont concrétisées par des processus d'analyse de traces composés d'une succession ordonnée d'opérations, implantées dans un outil d'analyse, appliquées sur des traces d'apprentissage. Ces processus d'analyse sont soumis à des contraintes qui sont soit liées aux contextes d'apprentissage, soit aux spécificités techniques des données et des outils d'analyses. Ces contraintes rendent le partage, la réutilisation et l'adaptation des processus complexes voire peu pertinents (Clow, 2012). Pour faciliter la capitalisation des processus d'analyse de traces, nous nous appuyons sur une approche narrative visant à structurer sémantiquement un processus d'analyse de traces d'apprentissage ainsi que les informations associées, décrivant le contexte d'apprentissage, caractérisant les traces, et justifiant les choix d'analyse effectués (Lebis et al., 2018). Cette démarche, fondée sur une ontologie, vise à rendre les analyses compréhensibles, la compréhensibilité étant l'une des propriétés nécessaires à la capitalisation. Une analyse est compréhensible si les différents aspects de l'analyse sont appréhendables par les acteurs concernés. Pour cela, il faut fournir des informations techniques, mais aussi des informations conceptuelles, comme les objectifs de l'analyse, les théories scientifiques utilisées, ou encore les choix d'implantation. Nous proposons ainsi de décrire un processus d'analyse par une succession d'étapes dans lesquelles des opérateurs d'analyse sont utilisés. Un *opérateur narré* représente le concept d'opération commun à des opérateurs similaires, lorsque sa sémantique est non ambigüe, et lorsque différentes informations sont intégrées et structurées. Nous définissons le comportement de ces opérateurs sur les variables à l'aide de différents *patrons*, eux aussi sémantiquement définis. L'analyste peut associer à l'ensemble du processus d'analyse, mais également à chaque étape de ce processus, des éléments narratifs, fondés sur un vocabulaire partagé, et permettant d'intégrer de manière structurée des informations liées au contexte de l'analyse et aux choix effectués. Cette explicitation par l'analyste de différents éléments décrivant une analyse représente un effort

conséquent. Cependant, le fait d'utiliser une ontologie et un vocabulaire contrôlé permet la congruence d'interprétations entre le système et l'utilisateur, et permet ainsi au système de rendre des services à la communauté des utilisateurs analystes. En effet, si un utilisateur exprime un besoin d'analyse en utilisant le vocabulaire contrôlé, le système peut retrouver des processus d'analyse existants et répondant au même besoin. Il peut également montrer à l'utilisateur dans quelle mesure ces processus existants ont été utilisés dans des contextes similaires au contexte du besoin exprimé. Il peut enfin lui expliquer quelle adaptation effectuer sur un processus d'analyse existant pour faciliter son application à un contexte similaire.

Exploration de traces pour la découverte interactive de connaissances. Transmute (Fuchs et Cordier, 2018) est une approche de découverte interactive de connaissances à partir de traces qui s'appuie sur la fouille de données pour mettre en évidence des régularités dans les traces sous la forme de sous-séquences d'événements appelées épisodes séquentiels. L'utilisateur, expert du domaine d'application, est impliqué pour sélectionner des épisodes *intéressants* et les interpréter pour construire un modèle du phénomène étudié. Les obsels composant la trace sont associés à des représentations choisies par l'utilisateur qui font sens pour lui, des symboles graphiques par exemple, faisant référence à des concepts connus de l'utilisateur (figure 2). Les épisodes sont représentés de la même façon et localisés dans la trace pour améliorer leur compréhension en contexte. Le système à base de traces fournit des possibilités de transformations dont la réécriture qui crée, à partir des épisodes sélectionnés, une nouvelle trace d'un niveau plus abstrait associé à son modèle. Le modèle du phénomène étudié qui a été construit à l'issue de l'interprétation est explicable car il est possible de naviguer vers ses éléments constitutifs dans la trace dont il est issu, et ceci de manière récursive.

FIG. 2 – *Extrait de l'interface de Transmute avec des traces issues d'un jeu sérieux.*

KATIE [11] (Fuchs, 2017) est une approche d'acquisition de connaissances qui vise à assister, en interaction avec l'utilisateur, le processus de modélisation et d'intégration des traces dans un système à base de traces, en détectant et corrigeant les erreurs résiduelles dans les données (données bruitées ou manquantes). À partir d'une trace fournie sous la forme d'un jeu de données brutes, KATIE extrait un modèle de trace [12] qui est proposé à l'utilisateur. Lorsque le modèle proposé ne correspond pas aux connaissances que l'utilisateur possède sur les données, ce dernier peut exprimer ses connaissances sous la forme de contraintes sur les données. Les

11. Knowledge Acquisition from Traces with Interactive Exploration
12. Le terme schéma de classes est souvent utilisé, quoique non équivalent. Ici il s'agit d'une hiérarchie des types d'obsels qui correspond à un modèle de connaissances de la trace.

données discordantes par rapport à ces contraintes sont extraites et montrées à l'utilisateur qui décide des actions à mener pour corriger les éventuelles erreurs : suppression de données bruitées ou ajout de données manquantes. Une fois les données modifiées en conséquence, KATIE réitère la construction du modèle de trace jusqu'à ce qu'un consensus soit obtenu avec l'utilisateur. Ce dernier peut alors interpréter les différents concepts proposés. Finalement, le modèle de trace est créé dans le système à base de traces et la trace y est ensuite enregistrée conformément à ce modèle. La génération du modèle de trace s'appuie sur l'analyse de concepts formels (Ganter et Wille, 2012) et les contraintes de l'utilisateur sont exprimées à l'aide d'implications de la forme : *tous les obsels qui possèdent l'attribut x possèdent également l'attribut y*. Le processus explore itérativement et alternativement l'*intent* et l'*extent* pour expliquer et rectifier les désaccords entre les connaissances de l'utilisateur et le modèle sous-jacent aux traces. Ces approches donnent à l'utilisateur un rôle central dans la construction de connaissances et l'explicabilité y est une caractéristique recherchée. L'utilisateur peut lui-même choisir la représentation visuelle des éléments de la trace, et c'est *via* leur manipulation interactive, que les règles sous-jacentes à la découverte de connaissances sont explicitées. Les modèles de connaissances obtenus après interprétation sont ainsi sémantiquement et explicitement reliés aux données et informations dont ils sont issus.

S'approprier ses traces d'interaction et négocier les indicateurs d'apprentissage associés : le prototype Trace-Me. Dans le cadre de la mission COAT [13] menée par le CNRS en 2014, une étude a été réalisée pour associer à un dispositif d'apprentissage MOOC, un dispositif d'assistance aux utilisateurs apprenants pour l'appropriation et le contrôle des fonctions de traçage, et d'élaboration d'indicateurs à partir des traces d'interaction. Cette étude a permis d'expérimenter une manière d'implanter deux « bonnes propriétés » dans le dispositif pour l'appropriation par l'utilisateur des régulations à l'œuvre, ce que nous défendons ici comme définitoire de l'explicabilité :

1. Appropriation par l'utilisateur du processus de traçage
   - démarrage, suspension, reprise et arrêt de traçage sous son contrôle ;
   - choix de ce qui doit être tracé sur son poste pour une activité donnée : les apprenants choisissaient souvent de tracer l'accès à des ressources autres que celles offertes par la plateforme MOOC ;
   - stockage dans un environnement privé ;
   - visualisation interactive en temps réel, avec un effet de réflexivité de l'activité dans le temps et dans l'espace :
     - de ses interactions vécues dont il a l'expérience action-perception ;
     - des actions de la plateforme, non vécues par lui mais révélées en proximité visuelle de ses propres interactions et fournissant des clés du fonctionnement interne ;
   - mise en forme configurable : symboles, couleurs, champs textes ...
2. Appropriation des logiques de régulation par des fonctionnalités d'exploration et d'intervention sur la sémantique des traces synthétisées dans les indicateurs :
   - accès à un outil d'analyse des traces selon des modèles configurables,
   - accès à un outil de gestion d'indicateurs partagés dans un *store*, permettant d'accéder aux modèles des indicateurs, et d'y déposer les siens propres, développés avec l'outil de conception utilisé par les concepteurs enseignants (backoffice). Les indicateurs sont

13. Connaissance Ouverte A Tou.te.s `https://projet.liris.cnrs.fr/coatcnrs/wiki/doku.php`

affichés dans l'environnement d'apprentissage des utilisateurs apprenants (frontoffice). L'apprenant a accès au backoffice et au frontoffice.

Ce dispositif a été expérimenté lors de formations MOOC et pendant une école d'été du CNRS. Les plateformes de MOOC sont en réalité peu utilisées comme espace de travail collectif. Les apprenants utilisent massivement les outils génériques du Web qu'ils connaissent pour travailler ensemble. L'articulation entre plateforme et autres outils pour une même activité de l'utilisateur n'est pas assurée en général. Trace-me offre la possibilité d'observer son activité d'apprentissage qu'elle soit médiée par la plateform MOOC ou non. L'utilisateur maîtrise son traçage propre et peut intervenir dans la régulation en proposant d'adapter ou de créer de nouveaux indicateurs, en discussion avec les utilisateurs concepteurs ou non.

Dans cet exemple, on voit que le processus d'explication exige une explicitation de la régulation de l'apprentissage (traces et modèles d'indicateurs) mais aussi des fonctions d'exploration et de reformulation par l'utilisateur lui-même dans un engagement dans l'effort d'explication. Des prototypes similaires sont décrits par Cram et al. (2007, 2008).

6 Discussion et perspectives

Dans son analyse de l'opacité des systèmes d'IA, Burrell (2016) en identifie trois formes : l'opacité liée à un effort délibéré de garder *secret* le fonctionnement du dispositif, celle liée au *manque de compréhension* par les utilisateurs, et celle liée à la *complexité* intrinsèque du dispositif. Cette dernière forme n'est pas limitée aux systèmes d'IA numérique, car elle ne provient pas uniquement du manque d'explicitation : la combinatoire d'exécution d'un algorithme non-trivial en rend son appréhension difficile, y compris pour un spécialiste.

L'opacité liée au secret sort quelque peu du cadre de cet article, puisque notre approche suppose que les capacités d'explicabilité sont recherchées par les concepteurs du dispositif. On peut cependant noter qu'un effort de rétro-conception, s'appuyant sur des traces d'intectactions collectées *via* une observation extérieure au dispositif (Ginon et al., 2013), reste possible pour rendre un tel système explicable « malgré lui ».

L'opacité liée au manque de compréhension est la raison pour laquelle nous considérons insuffisante (bien que nécessaire) l'explicitation des régulations à l'œuvre dans le dispositif. Cette dernière doit s'accompagner de mécanismes interactifs de négociation de sens, pour mettre en congruence l'interprétation personnelle de l'utilisateur avec l'intérprétation canonique prescrite par les concepteurs du système. En tant qu'elles capturent une « expérience » partagée entre l'humain et le dispositif, les traces d'interaction constituent un médium de choix pour alimenter cette négociation. Elles sont éminemment *ambivalentes*, en ce sens qu'elles peuvent avoir une signification différente pour le système et pour l'utilisateur, mais aussi pour différents utilisateurs (par exemple un apprenant et un enseignant, dans le contexte d'une application éducative). Nous avons proposé (Champin, 2017) un cadre théorique pour apréhender cette ambivalence et formaliser les congruences entre les différences interprétions pouvant coexister dans ce cadre. Sa mise en pratique concrète reste cependant à évaluer dans des travaux futurs.

Enfin, la notion de trace transformée, centrale dans le modèle proposé en section 4, vise précisément à répondre à l'opacité liée à la complexité. Elles permettent en effet, d'une part, de reformuler des traces de « bas niveau » en traces plus abstraites et synthétiques, et d'autre part, d'expliquer les obsels transformés par leur lien généalogique avec les obsels sous-jacents. Bien

sûr, ces transformations elles mêmes ont leur part de complexité. Il convient donc de veiller à ce que l'explication ne soit pas plus opaque que le système qu'elle cherche à expliquer.

7 Conclusion

Cet article a été l'occasion de rappeler l'origine de l'ambition des recherches en intelligence artificielle, cherchant à étudier les fonctions cognitives fondamentales, pour les comprendre et en tirer profit en concevant des dispositifs techniques qui s'en inspirent pour être efficaces dans le traitement de tâches complexes. Certains systèmes se voulaient *explicables* par construction puisque mimant le raisonnement humain pour la résolution de problèmes, avec des heuristiques explicites et un effort à montrer la rationalité du raisonnement par sa trace symbolique et vérifiable. D'autres systèmes étudiaient les structures qui se construisent à partir d'un apprentissage *automatique* sans formulation d'un raisonnement explicite. Nous soutenons que l'explicabilité nécessite l'explicite mais le dépasse pour exiger une capacité à rapprocher les interprétations que l'utilisateur peut élaborer en situation d'utilisation d'un dispositif technique numérique et les régulations à l'œuvre telles qu'elles déterminent le comportement dudit dispositif. Nous parlons de *congruences d'interprétations*. Nous proposons d'utiliser les traces d'interaction comme support commun à l'utilisateur et au dispositif technique numérique pour l'exploration des régulations s'exprimant au travers des motifs d'interaction, à différents niveaux d'abstraction. La transformation d'une trace peu explicite en une trace interprétée nécessite une transformation explicite qui décrit d'une certaine manière les règles d'interprétation correspondant à telle ou telle régulation à l'œuvre dans cette séquence d'interactions. Lorsque l'utilisateur n'est pas surpris par le comportement du dispositif technique numérique, c'est qu'il l'a compris, qu'il est maintenant le technicien de la fonction concernée au sens établi par Simondon (1989) pour exprimer cette maîtrise. Les illustrations montrent naturellement des cas d'usage des traces en situation réflexive pour les utilisateurs, mais aussi comment les concepteurs s'emparent de la notion de trace modélisée et des outils théoriques et techniques développés autour de cette notion pour préparer les conditions de l'explicabilité. Les perspectives montrent que le champ d'usage des traces modélisées s'élargit, se formalise et porte un potentiel fort pour les travaux à venir en matière d'explicabilité.

Références

Buchanan, J. M. et E. H. Shortlife (1984). *Rule Based Expert Systems : The MYCIN Experiments of the Stanford Heuristic Programming Project.* (Addison-Wesley ed.). Readin, MA.

Burrell, J. (2016). How the machine 'thinks' : Understanding opacity in machine learning algorithms. *Big Data & Society 3*(1), 12.

Champin, P.-A. (2017). *Empowering Ambivalence – Supporting multiple interpretations in knowledge-based systems*. HDR, Université Claude Bernard - Lyon I, Lyon, France.

Champin, P.-A., A. Mille, et Y. Prié (2013). Vers des traces numériques comme objets informatiques de premier niveau. *Intellectica* (59), 171–204.

Clow, D. (2012). The learning analytics cycle : closing the loop effectively. In *Proc. of 2nd International Conference on Learning Analytics and Knowledge*, pp. 134–138. ACM.

Cornuéjols, A., L. Miclet, et V. Barra (2018). *Apprentissage Artificiel. Deep learning, concepts et algorithmes (3rd Ed)* (Eyrolles ed.).

Cram, D., B. Fuchs, Y. Prié, et A. Mille (2008). An approach to user-centric context-aware assistance based on interaction traces. *MRC 2008, Modeling and Reasoning in Context*.

Cram, D., D. Jouvin, et A. Mille (2007). Visualizing Interaction Traces to improve Reflexivity in Synchronous Collaborative e-Learning Activities. In A. C. Limited (Ed.), *6th European Conference on e-Learning*, pp. 147–158.

Fuchs, B. (2017). Assister l'utilisateur à expliciter un modèle de trace avec l'analyse de concepts formels. In C. Roussey (Ed.), *IC 2017*, Caen, France, pp. 151–162.

Fuchs, B. et A. Cordier (2018). Interactive interpretation of serial episodes : experiments in musical analysis. In C. Faron-Zucker et C. Ghidini (Eds.), *EKAW-2018*, LNAI 11 313, Nancy, France, pp. 131–146. Springer.

Ganter, B. et R. Wille (2012). *Formal concept analysis : mathematical foundations*. Springer Science & Business Media.

Ginon, B., P.-A. Champin, et S. Jean-Daubias (2013). Collecting fine-grained use traces in any application without modifying it. In *workshop EXPPORT from the conference ICCBR*.

Hasan, R. (2014). *Predicting query performance and explaining results to assist Linked Data consumption*. Theses, Université Nice Sophia Antipolis.

Lebis, A., M. Lefevre, V. Luengo, et N. Guin (2018). Capitalisation of analysis processes : enabling reproducibility, openness and adaptability thanks to narration. In *Proc. of the 8th International Conference on Learning Analytics and Knowledge*, pp. 245–254. ACM.

Liquète, V., E. Delamotte, et F. Chapron (2012). L'éducation à l'information, aux tic et aux médias : le temps de la convergence ? *Études de communication 38*.

Mille, A. (2019). Vers des dispositifs techniques numériques orientés éthique ? *Intellectica* (70), 119–164.

Newell, A. (1982). The Knowledge Level. *Artificial Intelligence* (18), 87–127.

Newell, A., J. C. Shaw, et H. A. Simon (1959). Report on a general problem-solving program. In *Proceedings of the International Conference on Information Processing*, pp. 256–264.

Olah, C., A. Satyanarayan, I. Johnson, S. Carter, L. Schubert, K. Ye, et A. Mordvintsev (2018). The Building Blocks of Interpretability. *Distill 3*(3).

Simondon, G. (1989). *Du mode d'existence des objets techniques*. Philosophie. Aubier.

Summary

The three stages of Explainable AI: How explainability facilitates real world deployment of AI

Clodéric Mars*, Rémi Dès*, Matthieu Boussard*

*craft ai
8, rue de la Michodière, 75002 Paris
prenom@craft.ai
https://www.craft.ai

Résumé. L'intérêt pour l'IA explicable s'est récemment vu renouvelé, et nous pensons que ces approches permettent de faire une vraie différence dans le déploiement d'IA, particulièrement dans le monde de l'entreprise. Dans cet article nous introduisons un cadre permettant de catégoriser les niveaux d'explicabilité, leurs impacts dans l'opérationnalisation d'IA, et leurs prérequis.

1 Introduction

The explainability of AI has become a major concern for AI builders and users, especially in the enterprise world. As AIs have more and more impact on the daily operations of businesses, we see trust, acceptance, accountability and certifiability become requirements for any deployment at a large scale.

1.1 XAI

Explainable AI (XAI) as a field was popularized by the eponymous DARPA program launched in 2017, with the goal of creating a suite of machine learning techniques that produce more *"explainable"* models while maintaining a high level of learning performance, thus enabling human users to understand, trust and effectively manage the emerging generation of AI systems (Gunning, 2017).

Over the past 5 years, explainability has become a key part of the AI industry strategy for countries (Holdren et Smith, 2016; Villani et al., 2018) or research institutions (Braunschweig, 2016). It is also a strategic axis for companies, small (Mars, 2019; Guggiola et al., 2018) or large, for example through the publication of open source libraries (Microsoft, 2019; IBM, 2019) or dedicated services (Google). Major AI conferences dedicate workshops to this topic (e.g. First Workshop on Explainable Artificial Intelligence (XAI) IJCAI 2017). Specialized MOOC are being launched (Becker, 2019).

1.2 What is an explanation?

Explainable AI is about providing explanations regarding AI processes to stakeholders, it is therefore interesting to look at how people explain their decisions to each others. Specifically,

the design of XAI can benefit from the learnings of social sciences on explanation. In his thorough review, Miller (Miller, 2019) studied works from various branches of social sciences from philosophy to cognitive science and psychology.

The surveyed body of work tends to show that people seek to build a mental model of how decisions are made or how events occur, in order to anticipate them and reason about them. Explanations are a way to build such models much quicker than through observation only. Because mental models are inherently subjective, good explanations are biased towards the explainee to match their perspective and their preexisting knowledge. In the real world examples we describe below, we found that the work of understanding the point of view of the explainee is a major part of the design of explainable AIs.

Another major finding is that good explanations are contrastive. It is not about answering *"why has event E occurred?"* but rather *"why has event E occurred instead of another event C?"*. We found out that the capability to generate such *constrative* or *counterfactual* explanations is quite important in the deployed systems we describe in section 2.2.

Miller argues that Explainable AI as a field should be considered at the crossroad of Social Science, Human-Computer Interaction and Artificial Intelligence. Taking a more practical approach, in this article we will take the point of view of the people and systems interacting with AI systems, and study how explainability impacts these interactions in terms of features, acceptance and capacity to be deployed.

2 How XAI makes a difference

In order to study the impact XAI makes on AI projects we are categorizing effects in three stages described in Fig 1. Higher stages require higher levels of explainability and have more impact on the resulting AIs. We take the point of view of the industrial world, and look at how explainability can make a difference in the deployment and application of AI.

This work is based on the experience we gathered working and discussing with our customers, partners and community, as a provider of machine learning solutions. Examples are focused on systems based on Machine Learning but the proposed three stages are relevant to any kind of AI.

2.1 Stage 1: Explainable building process

In any organisation, just like any IT project, a project leveraging AI aims to have an impact on the daily job of some people. Its goal might even directly be to automate part of worker's job or to help them deliver value they could not before. Especially when AI is involved, affected users can be wary of the new system. In particular they may feel threatened by the automation of some of their tasks, or may not believe that a simple *computer program* can execute complex tasks correctly. A recommandable method to address those concerns is to involve them in the building of the AI. This is where explainability plays a big role.

In this context, traditional quality metrics such as confusion matrices, r2, RMSE, MAE, etc. are not sufficient to get the future AI user's trust, since they want to know more about the *why* than about the raw results. Visualization is the first go-to technique. Simply plotting the output against context variables is a good way to get a *feel* for how an AI performs over the target domain when dimensionality is low. Interactive simulations can help explore the

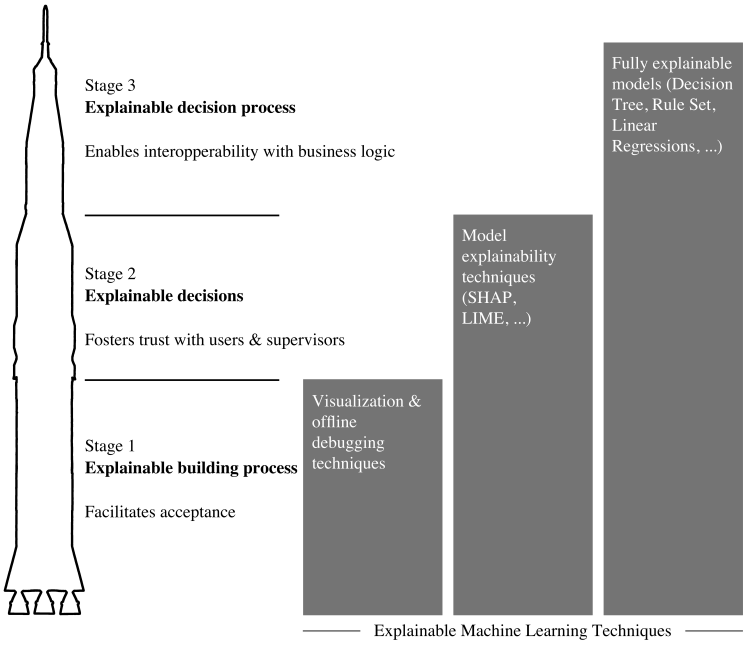


FIG. 1 – *The three stages of AI explainability*

domain to experience how the AI will react. Beyond these techniques which are applicable to any *black box* computations, more advanced techniques open the hood and make the structure of the AI itself inspectable.

In the following sections describing the subsequent stages, we will talk about techniques able to work while the AIs are *live*, processing production data, at production speed. These techniques are also well suited for stage 1, where the inspection is offline, with less data and runtime constraints.

Debugging tools that were initially designed for data scientists can also be leveraged for other stakeholders. AIs powered by neural networks can be inspected by visualizing how intermediate layers *react* to different input, Tensorflow Playground (Smilkov et Carter) or ConvnetJS (Karpathy, 2014) are good examples of this approach. On images, the computation of *saliency maps* can also help to convey which parts of the image are considered by the network to make its prediction (Simonyan et al., 2013). This technique led to the identification of the infamous *husky vs wolf* issue in which a wolf is primarily identified by the presence of snow in the picture (Ribeiro et al., 2016b). Tools like Seq2Seq Vis (Strobelt et al., 2018) bring the same kind of debugging capabilities to natural language focused neural networks. This shows that even neural networks, which are considered black boxes, can be at least partly explained offline to the non-technical AI project stakeholder by using the right tools.

While the initial goal of explaining why the AI works the way it does is to ease its adoption, explainability also increases the involvement of potential users by letting them achieve a deeper understanding. As a result they can assist in its development, ensuring that the AI solves an

actual problem, and provide valuable feedback on specific behaviors of the AI: instead of providing knowledge upfront, it is always easier to *react* to what you see the AI doing and why it does it. In many cases, domain experts can easily help if they have an understanding of why the AI makes decisions: sensors having an undocumented validity domain, well-known contexts leading to corrupted data, spurious correlations because of a missing data sources, etc.

The first stage of explainability is about helping create a multi disciplinary team of experts in their respective fields who understand the AI they are building. Offline explainability techniques are key to the acceptance of the future AI and create opportunities to build a better system.

2.2 Stage 2: Explainable decisions

Trust in a system is key, especially in an enterprise tool that has an impact on day to day business. Trust makes the difference between a system that is *"micro managed"* by its users or supervisors, and a system that can enjoy a larger autonomy. The more management a system needs, the more manpower it requires and therefore the less value it has.

Trust is built when a system is not surprising, when it behaves according to our mental model. A system whose limits are understood by its users is arguably more valuable than a more accurate system whose results are considered unreliable. As discussed in section 1.2, explanations are a good way to accelerate the construction of this mental model. That is where the capacity to explain the AIs' decisions has an impact. That is the second stage of explainable AI.

Stage 1 explainability does not have the same impact: most users or supervisors of AIs did not have the chance to participate in their inception, and in more and more cases, AIs can evolve over time. Furthermore, the ability to access explanations of past AI decisions can help pinpoint root causes and generally provide traceability.

The ability to provide explanations to any AI decision is an active field of innovation with methods such as TreeInterpreter (Saabas, 2014), LIME (Ribeiro et al., 2016a) or SHAP (Lundberg et Lee, 2017). Given a predictive model and a prediction, these methods aim at providing a local explanation for the prediction. This explanation takes the form of linear factors that can be applied to input features to reach the predicted results, thus giving an idea of the local feature importance and behavior of the model. The computed feature factors can also be used to generated counterfactual examples and give an idea of the trend of the predicted value given changes in the input features.

An interesting property of this class of algorithm is that they can work using a feature set that is different from the actual feature set used by the model. It is therefore possible to adapt the explanation by making it more comprehensible to the explainee, independently from the features that yield the best predictions. This additional feature engineering step is not without risk, as it can be used to convince explainees to blindly trust said AI, by presenting a deceptive approximation instead of bringing more transparency (Denis et Varenne, 2019).

A good example of SHAP usage can be found in the banking fraud detection solution (Mars, 2019) provided by the Bleckwen company. One key part of the solution is a predictive model, trained on labelled datasets containing fraudulent and non-fraudulent transactions. This model computes a score for each transactions. Transactions having a score above a certain threshold are reviewed by a human expert to confirm their fraudulent nature. One of their customers' requirements is to get explanations for every score. They chose to use non-explainable

gradient boosting techniques for the model on a range of complex features. The local explanation is computed by SHAP on a range of features they designed with their end users to make them completely understandable to them.

Another example of stage 2 explainable AI is how Dalkia uses machine learning as a part of their energy management dashboard. Here, decision trees are used to predict an energy diagnosis based on labelled data streams. Predictions are used as diagnosis recommendations in the energy managers' dashboard, and explanations are extracted from the decision tree as a set of rules that were applied (craft ai, 2018). What's really interesting in this example is that without explanations alongside the recommendations this AI would not have any value. At its core, the goal of the system is to help energy managers handle more data points. Without an explanation, when provided with a prediction, energy managers would need to investigate the raw data in order to confirm or contradict it. They would end up doing the same amount of work as without explanations. When an explanation is provided, this counter investigation is only needed when the energy manager disagrees with it. Here, explanations are needed for the business value of the AI.

2.3 Stage 3: Explainable decision process

Stages 1 and 2 are about helping humans create a mental model of how AIs operate. This enables humans to *"reason"* about the way AIs work critically, and decide when to trust them and accept their outputs, predictions or recommendations. To scale this up to many AIs and over time, you need to define business logic that will apply the same *"reasoning"* automatically. Stage 3 is about enabling interoperability between AIs and other pieces of software, especially software that uses business logic.

When discussing AI, and especially models generated through machine learning, we often talk about the underlying concepts they capture, for example convolutional neural networks are able to recognize visual patterns and build upon these lower level *"concepts"* in their predictions. AIs that can explain those lower level building blocks, make them inspectable to business logic, reach stage 3. Such AIs ultimately act as a knowledge base of the behavior they model.

Stage 3 explainability makes a difference especially when a lot of instances of evolving AIs need to be supervised by business logic, for example in a context of continuous certifiability or collaborative automation between machine learning based AIs and business rules.

This level of explainability requires fully explainable AI. Machine learning techniques such as linear regressions or decision tree learning (Quinlan, 1993) can reach such levels. Another approach is to approximate a more *"black box"* model with a more explainable model, for example RuleFit is able to learn a minimal ensemble of rules from a tree ensemble method such as Random Forest (Friedman et al., 2008).

An interesting example of level 3 explainability is Total Direct Energie's energy coaching feature that is part of their customer-facing mobile application. It generates personalized messages for each customer (craft ai, 2019). At its core, the system is made of a machine learning-based energy consumption predictive model, and a business expertise-based message generation and selection module. The predictive model is made of individual regression trees, each updated continuously from the data of a single household. The message generation module is generic for all users, and uses the model's explanations and predictions as input data to select and personalize each message. So the predictive models provide an understanding of

the household's energy consumption behavior, which is automatically processed to generate personalized messages.

When presented with a visual explanation of a decision process, people tend to navigate through its structure to understand the process. Stage 3 is about letting software programs, other AIs, do the same thing, thus unlocking a wealth of additional use cases.

3 Challenges

While there are already deployed AIs covering these three stages, there are still challenges ahead before explainable AI can be generalized.

3.1 Evaluating explanations

In the previous sections we discussed how certain techniques bring more or less explainability, however we did not discuss how we can make such an assessment.

Ad-hoc experiments or KPI can be used. For example the D-Edge company, which provides pricing recommendations to hotel managers among other services, measures whether explained recommendations are accepted. Every recommendation is accompanied by a natural language explanation. Managers can accept and apply the recommendation to their pricing or discard it. As presented during a round table focused on XAI (Mars, 2019), they consider the proportion of accepted recommendations as a proxy measure for the quality of their explanation. We believe that this makes sense, as hotel managers need to be convinced to make such an impactful change to their business.

In the general case, other proxy measures can be used, such as the number of rules, nodes or input variables considered in an explanation or explainable model. However these lack generality: how can the explainability of a linear regression and of a regression tree be compared? They also lack an experimental, measurable ground truth: for example we do not know if humans find that the explainability provided by LIME grows exponentially or linearly with the number of features involved. Furthermore, as discussed in section 1.2, what constitutes a good or a bad explanation depends on the recipient of the explanation and their own cognitive biases (Denis et Varenne, 2019). This poses an additional challenge to this evaluation. There is a lack of a systemic framework or objective criteria to evaluate the explanations provided by AIs (Weller, 2017).

3.2 Improving the performances of XAI

The AI community generally considers that the more explainability you gain, the less predictive performance you can achieve, especially in Machine Learning. Overcoming this is a primary goal of the XAI field, and in particular it is the main goal of the DARPA XAI program (Gunning, 2017). Several opportunities have been identified to achieve this objective, the most promising ones being to create hybrid AI combining different approaches. One idea is to *push* high-performance but unexplainable algorithms to the edges, around an explainable core. For example in cat image recognition, a deep neural networks would identify low level details like whiskers and pointy ears, while decision trees or bayesian models would associate the presence of both whiskers and pointy ears to a cat in an explainable fashion. Another idea is to adapt

Machine Learning algorithms to work from existing expert-built symbolic representations of physical models to leverage existing knowledge, instead of having to relearn and embed it. This field is relatively new, and comes as a stark departure from the deep learning trend of the past few years.

4 Conclusion

In this paper we structured in three stages the impact that explainability can have on AI applications deployed in the *"real world"*. Those 3 stages provide a simple framework to quickly identify the need for explainability in a AI powered project. Stage 1 is about leveraging explainability to improve the adoption and performance of AI applications. Stage 2 is about explaining every AI decisions to build trust with their users and supervisors. Stage 3 is about enabling the interoperability of AI systems with each other and other software, thus unlocking new and richer use cases.

Because we focused on what explainability enables in AI, we did not discuss regulation. However it is important to note that initiatives such as the European GDPR pave the way for a *"right to explanation"* which will require, at least in some cases, a stage 2 requirement (Burt, 2017). We strongly believe that stage 2 explainability is a key to actually operationalize enterprise AI because it not only offers stronger guarantees in terms of data governance, but also facilitates involvement and support from users and domain experts impacted by such AI.

Far from being just a constraint on AI design, explainability helps develop better and richer AIs.

References

Becker, D. (2019). Machine learning explainability.

Braunschweig, B. (2016). Intelligence artificielle, les défis actuels et l'actions d'inria. Technical report, INRIA.

Burt, A. (2017). Is there a 'right to explanation' for machine learning in the gdpr?

craft ai (2018). How the deployment of an explainable ai solution improves energy performance management at dalkia.

craft ai (2019). How total direct energie applies explainable ai to its virtual assistant.

Denis, C. et F. Varenne (2019). Interpretability and explicability for machine learning: between descriptive models, predictive models and causal models. A necessary epistemological clarification. In *National (French) Conference on Artificial Intelligence (CNIA) - Artificial Intelligence Platform (PFIA)*, Toulouse, France, pp. 60–68.

Friedman, J. H., B. E. Popescu, et al. (2008). Predictive learning via rule ensembles. *The Annals of Applied Statistics 2*(3), 916–954.

Google. Google explainable ai.

Guggiola, A., J.-M. Schertzer, A. Hoff, C. Ledoux, S. Monnier, O. Wautier, et H. Stalla-Bourdillion (2018). Ia explique toi ! Technical report, Quantmetry.

Gunning, D. (2017). Xai, program update november 2017. Technical report, DARPA.

Holdren, J. P. et M. Smith (2016). Preparing for the future of artificial intelligence. Technical report, Executive Office of the President National Science and Technology Council Committee on Technology. Washington, DC.

IBM (2019). Ai explainability 360. IBM.

Karpathy, A. (2014). Convnetjs: Deep learning in your browser.

Lundberg, S. M. et S.-I. Lee (2017). A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*, pp. 4765–4774.

Mars, C. (2019). Explainable ai, a game changer for ai in production - ai night 2019 workshop.

Microsoft (2019). Interpretml.

Miller, T. (2019). Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence 267*, 1–38.

Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Ribeiro, M. T., S. Singh, et C. Guestrin (2016a). Model-agnostic interpretability of machine learning. *arXiv preprint arXiv:1606.05386*.

Ribeiro, M. T., S. Singh, et C. Guestrin (2016b). Why should i trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1135–1144. ACM.

Saabas, A. (2014). Interpreting random forests.

Simonyan, K., A. Vedaldi, et A. Zisserman (2013). Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*.

Smilkov, D. et S. Carter. Tensorflow playground. Google.

Strobelt, H., S. Gehrmann, M. Behrisch, A. Perer, H. Pfister, et A. M. Rush (2018). Debugging sequence-to-sequence models with seq2seq-vis. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pp. 368–370.

Villani, C., Y. Bonnet, C. Berthet, F. Levin, M. Schoenauer, A. C. Cornut, et B. Rondepierre (2018). Donner un sens à l'intelligence artificielle: pour une stratégie nationale et européenne. Technical report, Conseil national du numérique.

Weller, A. (2017). Challenges for transparency. *arXiv preprint arXiv:1708.01870*.

Summary

Explainable AI has recently seen a renewed interest. We believe these techniques make a true difference when it comes to deploying AIs, especially in the entreprise world. In this article we introduce a framework categorizing explainability levels, their impact on operationalized AI and their requirements.

Interface de Recueil de Données Imparfaites pour le CrowdSourcing

Constance Thierry*, Géry Casiez**, Jean-Christophe Dubois*, Yolande Le Gall*, Sylvain Malacria**, Arnaud Martin*, Thomas Pietrzak**, Pierrick Uro**

*Univ Rennes, CNRS, IRISA, DRUID
prenom.nom@irisa.fr,
http://www-druid.irisa.fr
**Inria Lille - Nord Europe, CRIStAL - UMR 918
prenom.nom@inria.fr,
https://loki.lille.inria.fr/

Résumé. Le *crowdsourcing* se définit par l'externalisation de tâches à une foule de contributeurs. La foule, très diversifiée sur ces plateformes, inclut des contributeurs de niveaux de qualification variables et qui réalisent la tâche de manière plus ou moins consciencieuse. Cette diversité de profils induit des contributions de qualité inégale. Afin de palier ce problème nous étudions une interface utilisateur offrant aux contributeurs une plus grande capacité d'expression dans leurs réponses. Elle permet aux contributeurs de fournir une réponse imprécise en cas d'hésitation ainsi que de renseigner leur confiance dans leurs réponses. Cette interface a été déployée en demandant aux utilisateurs de répondre à des questions de difficulté variable et dont les réponses sont connues. Deux plages de difficulté ont été testées et nos résultats suggèrent que la certitude reportée par les utilisateurs est bien fonction de la difficulté absolue de la question, et non pas répartie de manière relative sur la plage de difficulté.

1 Introduction

Aujourd'hui encore, certaines activités ne peuvent être réalisées par ordinateur car elles sont trop coûteuses en temps ou nécessitent une expertise humaine. Le *crowdsourcing* permet d'externaliser ces tâches sur des plateformes dédiées afin qu'elles soient accomplies par une multitude d'utilisateurs, appelés contributeurs. Les données recueillies issues de sources hétérogènes sont ensuite fusionnées pour en extraire une information fiable et permettre une prise de décision. Cette pratique offre aux employeurs l'avantage de traiter une tâche dans de courts délais et avec un coût réduit. Cependant une problématique du domaine est d'obtenir des données fiables et pertinentes. En effet, les contributions humaines de qualité inégale, comportent des imperfections inhérentes à toute évaluation subjective, parfois amplifiées par un manque d'expertise ou de sérieux, mais aussi liées au mode de recueil des données. Ainsi, alors que les contributeurs sont fréquemment confrontés à des situations d'indécision, les possibilités d'expression des interfaces de recueil d'informations ne permettent pas de formuler finement

les réponses ni d'indiquer la validité accordée. Nos travaux ont pour objectif de proposer une interface dédiée au *crowdsourcing* permettant aux contributeurs d'exprimer des réponses plus riches comportant des imperfections exploitables dans le processus de décision à l'aide de théories de l'incertain. A l'instar de Koulougli et al. (2016), Ouni et al. (2017), Abassi et Boukhris (2018), Ben Rjab et al. (2016) nous avons opté pour une modélisation fondée sur les fonctions de croyance.

La section 2 présente l'interface proposée et le protocole expérimental suivi, la section 3 présente une analyse des résultats obtenus. Puis, la section 4 compare une modélisation des réponses par la théorie des fonctions de croyance avec une modélisation probabiliste et un vote majoritaire. Enfin nous concluons cette étude et présentons nos perspectives de travail.

2 Interface et protocole expérimental

L'interface étudiée a pour objectif d'offrir au contributeur la possibilité d'exprimer une réponse imprécise, en lui permettant de donner plusieurs choix en cas d'hésitation, mais aussi incertaine en attribuant un niveau de confiance en sa réponse. Afin de valider cette interface, 3 phases expérimentales sont considérées afin d'analyser distinctement le comportement du contributeur confronté à des situations d'incertitude puis d'indécision et enfin aux deux simultanément. Nous aborderons dans cet article uniquement la première phase expérimentale visant à valider qu'il y a bien une corrélation entre la facilité de réalisation d'une tâche et la certitude exprimée par le contributeur en s'appuyant sur une courbe psychométrique. Cette campagne de *crowdsourcing* a été réalisée sur la plateforme Figure Eight [1].

La tâche repose sur la perception visuelle, ce qui nous permet d'avoir une vérité terrain, de contrôler de manière continue la difficulté de la tâche liée à la différence de longueur entre les segments et de ne pas nécessiter de qualification spécifique de la part du contributeur. Dans cette expérimentation dont l'interface est présentée figure 1, il est demandé au contributeur de désigner le plus grand segment parmi un ensemble de 5 segments, en renseignant la certitude dans sa réponse. La réponse correcte est inclue dans les 5 propositions, ce qui donne un taux de chance de $20\ \%$. La *position* de la bonne réponse pouvant influencer les résultats, chaque question est dupliquée pour les 5 positions possibles.

L'échelle de certitude présentée au contributeur est la suivante : totalement incertain (0), incertain (1), peu incertain (2), ni certain ni incertain (3), peu certain (4), certain (5), totalement certain (6). Il est notifié au contributeur au début de la campagne qu'il a le droit d'être incertain dans sa réponse et que ceci n'aura pas d'impact lors de sa rémunération.

La difficulté de cette tâche suit une loi de Weber, c'est-à-dire que le rapport entre le seuil de perception Δ_I et la valeur témoin I est une constante. La longueur témoin a été déterminée empiriquement à $40\ mm$. La difficulté variait selon la valeur de Δ entre la bonne réponse et les bâtons témoins. Nous avons mené des études empiriques en testant notre interface en laboratoire afin de déterminer Δ_I.

Lors de ces études préliminaires, nous avons observé que les participants estimaient leur certitude de manière relative, c'est-à-dire qu'ils donnaient une forte certitude pour la valeur de Δ la plus élevée, et que les valeurs de certitude intermédiaires étaient réparties entre les valeurs de Δ extrêmes (Figure 3, courbes Labo). Néanmoins les participants ne parviennent

1. https ://www.figure-eight.com

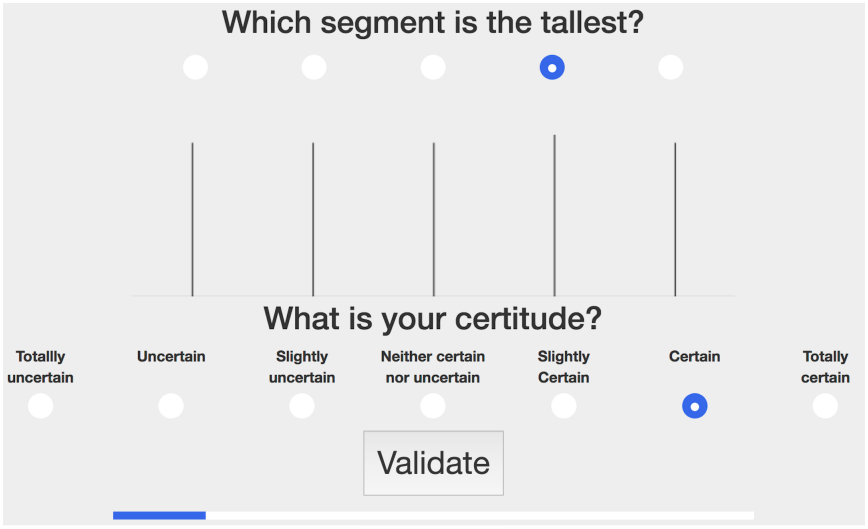


FIG. 1 – *Interface proposée aux contributeurs.*

pas à assigner les valeurs extrêmes de certitude à leurs réponses, par exemple si ces certitudes ont des valeurs de 0 et 5, alors les participants indiqueront généralement leur certitude entre 1 et 4. Nous avons donc choisi d'expérimenter deux séries Δ_0 et Δ_1 afin de vérifier cette hypothèse. Ces deux séries ont été effectuées par des participants différents afin d'éviter un effet d'apprentissage qui pourrait biaiser l'expérience. La valeur de Δ la plus basse est de 0. Le taux d'erreur pour cette valeur devra être proche du taux de chance. La valeur de Δ maximale est choisie afin d'être clairement identifiable selon nos études préliminaires. Les deux séries comportaient 6 valeurs, un compromis entre précision de mesure et durée de l'expérience. Les séries contiennent plus de valeurs autour du seuil de perception obtenu lors des études préliminaires ($300\ \mu m \leq \Delta_I \leq 600\ \mu m$) :

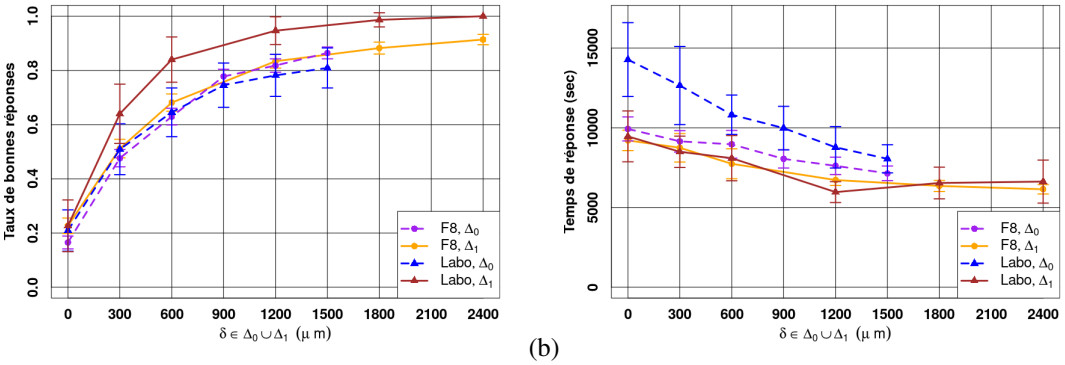
$$\Delta_0 = \{0, 300, 600, 900, 1200, 1500\}$$

$$\Delta_1 = \{0, 300, 600, 1200, 1800, 2400\}$$

Le protocole expérimental est donc 5 positions $\times$ 6 valeurs $\times$ 3 répétitions = 90 essais par participant. La série de valeurs est un facteur intra-sujet, balancé entre les utilisateurs. Il était de plus demandé aux participants d'installer le logiciel *pointing server* [2] afin de détecter la taille de l'écran du contributeur. Au total, 118 contributeurs ont réalisé la campagne de *crowdsourcing*, 63 pour Δ_0 et 55 pour Δ_1, la base de données des réponses considérées contient ainsi 5670 contributions pour Δ_0 et 4950 pour Δ_1. L'étude préliminaire réalisée en laboratoire incluant 6 contributeurs pour Δ_0 et 5 pour Δ_1 est également étudiée par la suite.

En fin d'expérimentation, plusieurs questions permettant d'avoir un retour sur l'expérience ont été introduites. Elles sont détaillées dans la section présentant les résultats.

2. `https://github.com/INRIA/libpointing`



(a) (b)

FIG. 2 – *Taux de bonnes réponses (a) et temps de réponse (b) des contributeurs.*

3 Analyse des résultats

Durant la campagne de crowdsourcing 118 contributeurs ont réalisé intégralement les tests. 20 contributeurs supplémentaires ont débuté l'expérimentation mais ne l'ont pas menée à terme. La figure 2(a) représente le taux de bonnes réponses de l'ensemble des contributeurs pour chaque degré de difficulté δ. Plus δ est grand, plus la différence de taille entre le plus grand segment et les autres est importante et donc plus la tâche est facile pour le contributeur. Conformément à ce qui est attendu, le taux de bonnes réponses des contributeurs augmente bien avec δ. De plus, les taux de bonnes réponses pour Δ_0 et Δ_1 sont très proches pour les données de la campagne de crowdsourcing et celles de laboratoire , ce qui montre que la difficulté de la tâche est bien perçue par le contributeur quelque soit l'ensemble de difficulté Δ choisi. —— Nous déterminons à cet effet la courbe psychométrique des utilisateurs présentant la distribution des certitudes renseignées selon une difficulté croissante de la tâche. ——

Considérons maintenant le temps de réponse du contributeur à la tâche. La figure 2(b) montre une diminution du temps de réponse moyen des contributeurs avec l'augmentation de δ. Ainsi, plus la tâche est facile, moins les contributeurs ont besoin de temps pour répondre car ils perçoivent rapidement quel est le segment le plus grand. Une autre constatation pour cette figure, est l'écart de temps de réponse entre les contributeurs. Les contributeurs ayant réalisé l'expérience avec Δ_0 ont eu besoin de davantage de temps pour répondre en comparaison aux contributeurs ayant réalisé l'expérience pour Δ_1. Cette différence de temps peut s'expliquer par le fait que l'écart des valeurs de $\delta \in \Delta_1$ est plus important. Par conséquent, le segment le plus long est plus rapidement perceptible pour les grandes valeurs de δ et induit davantage de rapidité pour les plus faibles valeurs. La difficulté générale de la campagne a donc un fort impact sur le temps de réponse des contributeurs sur ces plateformes.

Nous observons (figure 3) qu'en moyenne, les contributeurs de la plateforme de crowdsourcing FigureEight sont assez certains de leurs réponses, et ce même lorsque la tâche est la plus difficile et que tous les segments sont de longueurs identiques ($\delta = 0$). De manière positive, la diminution de la difficulté de la tâche avec l'augmentation de δ s'accompagne d'une

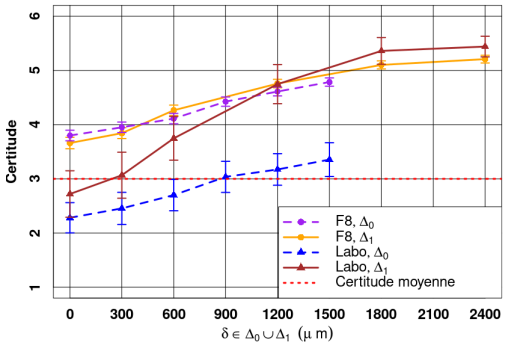


FIG. 3 – **Certitude moyenne** *des contributeurs en fonction de la difficulté de la tâche.*

augmentation de la certitude des contributeurs. Néanmoins, une diminution plus importante de la certitude lorsque la difficulté de la tâche augmente était attendue. En effet, lorsque tous les segments sont de taille égale ($\delta = 0$), il n'est pas possible pour le contributeur de sélectionner le bon segment. Dans ces conditions, le contributeur est alors contraint de choisir un segment au hasard, il devrait alors spécifier qu'il est totalement incertain (0). L'évolution de la certitude d'après la difficulté des questions pour les données de laboratoire avec Δ_1 est davantage en accord avec nos attentes. Cette difficulté de la part du contributeur de la plateforme de crowdsourcing de spécifier qu'il est incertain peut être due au fait qu'il n'ose pas affirmer cette incertitude de peur que son travail soit déprécié même si dans cette expérience, une contribution n'est pas dévalorisée en raison d'une forte incertitude. Au contraire, il est attendu de la part du contributeur que sa certitude en sa réponse varie suivant la difficulté de la question. Ainsi, un contributeur dont la certitude est constante est moins crédible qu'un contributeur qui reconnaît que sa certitude change suivant la difficulté de la question.

Ce dernier point est abordé dans la section suivante.

Un autre fait intéressant est l'évolution du taux de bonnes réponses des contributeurs au cours de la tâche. Afin de modéliser cette évolution le taux de bonnes réponses est mesuré pour une question q de la manière suivante :

$$TBR(q) = \sum_{c \in C} \sum_{i \in [1,q]} \frac{\text{isGoodAnswer}_{c,i}}{q * |C|}$$

Dans l'équation ci-dessus, C est l'ensemble des contributeurs ayant réalisé la campagne, et le booléen $\text{isGoodAnswer}_{c,i}$ est égal à 1 si le contributeur c a renseigné la bonne réponse à la question i. La figure 4 est la représentation graphique de l'évolution du taux de bonnes réponses des contributeurs au cours de la campagne pour Δ_0 et Δ_1. Pour les contributeurs ayant réalisé la campagne avec Δ_0, ce taux diminue faiblement au cours de l'expérience. Il semblerait que les contributeurs soient progressivement moins attentifs. Cette diminution de l'attention du contributeur peut s'expliquer par le fait que la campagne utilisant Δ_0 est plus complexe que celle avec Δ_1 et que la tâche devient fastidieuse au fur et à mesure de l'expé-

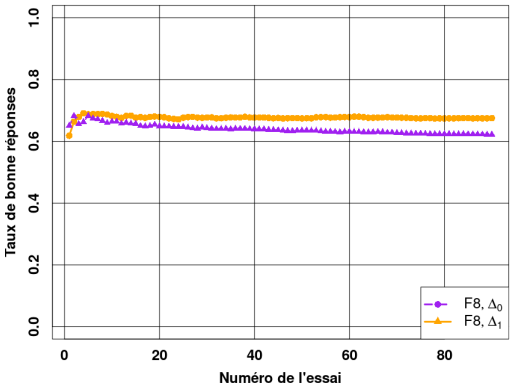


FIG. 4 – *Taux de bonnes réponses des contributeurs au cours de la tâche pour Δ_0 et Δ_1.*

rience. Pour les contributeurs ayant réalisé la campagne de crowdsourcing pour Δ_1, le taux de bonnes réponses varie un peu sur les premières questions puis est plus constant. Cette variation du taux sur les premières réponses est associé au temps d'apprentissage du contributeur, ce temps d'apprentissage est aussi visible pour la courbe associée à Δ_0 pour les premières questions.

Une fois la campagne réalisée par les contributeurs sur *Figure Eight*, il leur était demandé de répondre à un dernier questionnaire afin d'avoir leurs retours sur cette expérience. Les questions posées et réponses proposées sont les suivantes :

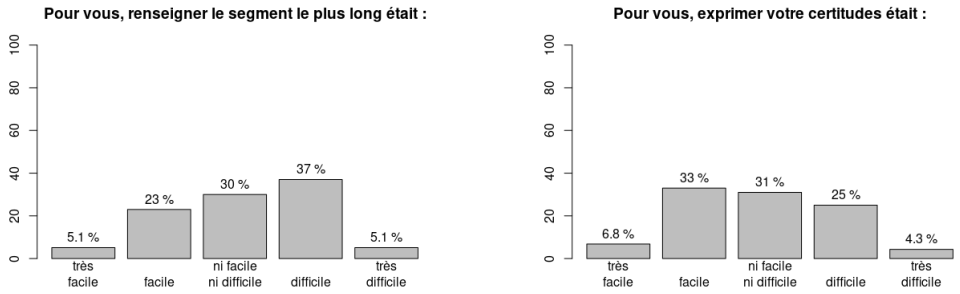
— (q1) Pour vous, renseigner le segment le plus long était :
Très facile, facile, ni facile ni difficile, difficile, très difficile.

— (q2) Pour vous, exprimer votre certitude était :
Très facile, facile, ni facile ni difficile, difficile, très difficile.

— (q3) Avez-vous hésité avant de répondre ?
Presque toujours, souvent, occasionnellement, rarement, jamais.

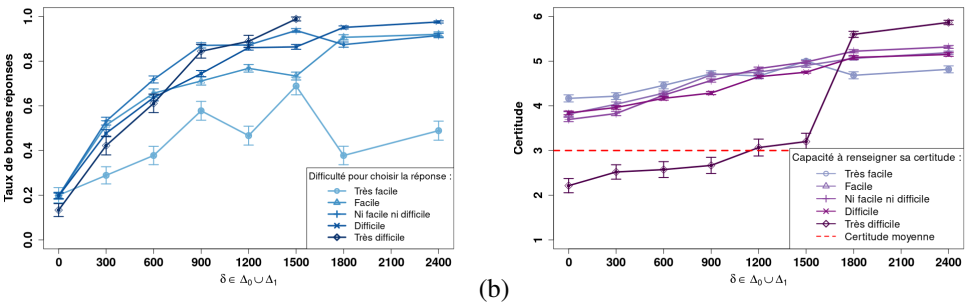
— (q4) Dans le cas où vous hésitez souvent, quel en est la cause ?
La tâche devient fastidieuse, vous avez un doute entre plusieurs segments, autre (à préciser).

Les pourcentages de contributeurs ayant répondu aux questions (q1) et (q2) sont illustrés figure 5, nous revenons sur ces résultats plus tard. A la question (q3), 15% des contributeurs ont répondu qu'ils hésitaient presque toujours, 25% souvent, 31% occasionnellement, 17% rarement et 13% considèrent ne jamais avoir hésité. Or, parmi l'ensemble des questions, à 5 reprises les segments étaient tous de même taille et il était donc impossible de ne jamais hésiter sur la réponse à renseigner. Pour la question (q4) seules les réponses des contributeurs qui admettent avoir douté sont considérées, soit 87% de la foule. La majorité de ces contributeurs (84%) reconnaissent que leur hésitation est liée au fait que la tâche devient fastidieuse,alors



(q1) (q2)

FIG. 5 – *Pourcentage de contributeurs d'après leurs réponses aux questions (q1) et (q2).*



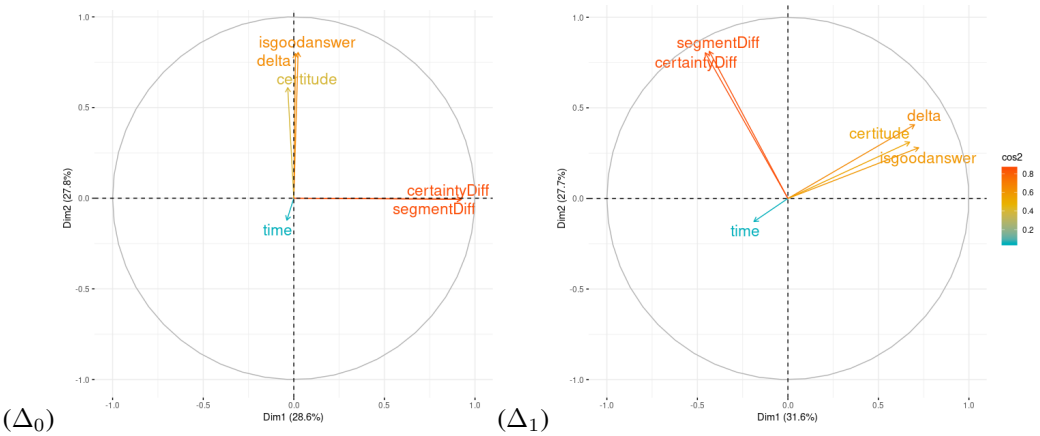
(a) (b)

FIG. 6 – *Taux de bonnes réponses et certitude pour 5 groupes de contributeurs.*

que les 16% restants indiquent que cette hésitation est due à la taille des segments.

La figure 6(a) est une représentation graphique du taux de bonnes réponses en fonction de la difficulté δ pour cinq groupes de contributeurs. Ces groupes sont définis d'après la réponse du contributeur à la question (q1). Le pourcentage de contributeurs appartenant à chaque groupe est donné par la figure 5(q1). Une part importante des contributeurs (37%) précise qu'il leur était difficile de renseigner parmi les 5 segments lequel était le plus long. Le groupe de contributeurs ayant spécifié qu'il était très facile de déterminer le segment le plus long est celui possédant le taux de bonnes réponses le plus faible. Plusieurs hypothèses peuvent être avancées telles qu'une mauvaise compréhension de l'expérience, ou encore un manque de sérieux qui les ont conduits à répondre aléatoirement à la campagne. Notons également que le groupe de contributeurs (5.1%) ayant reconnu que la tâche était très difficile est uniquement constitué de participants ayant réalisé la campagne pour Δ_0 ce qui confirme que Δ_0 est perçu comme plus difficile que Δ_1.

Considérons maintenant l'évolution de la certitude moyenne de groupes de contributeurs constitués d'après leur réponse à la question (q2). Le pourcentage de contributeurs apparte-



(Δ_0) (Δ_1)

FIG. 7 – *ACP : premier plan des variables.*

nant à chaque groupe est donné par la figure 5(q2). Le groupe comportant le pourcentage le plus élevé de contributeurs (33%) est composé d'internautes ayant indiqué que renseigner leur certitude en leur réponse était facile. Néanmoins en observant sur la figure 6(b) la courbe associée à ce groupe, nous constatons que leur certitude varie peu avec la difficulté de la tâche. Les contributeurs considérant qu'il leur était très facile de spécifier leur difficulté sont pourtant ceux pour lesquels l'évolution de la certitude est la plus faible. Leur perception de la difficulté de la tâche n'est donc pas corrélée avec le niveau de confiance renseigné pour leurs réponses. Par ailleurs, ce sont les contributeurs pour lesquels il était très difficile de donner leur certitude qui ont fait évoluer ce niveau de certitude de façon la plus importante en fonction de la difficulté. En effet pour les questions les plus difficiles leur certitude moyenne est assez faible, et cette dernière devient réellement importante pour les questions les plus faciles ce qui était attendu des contributeurs.

Pour clore cette section, une analyse en composantes principales (ACP) a été réalisée pour Δ_0 et Δ_1 en considérant les paramètres suivants :

— delta : la difficulté δ de la question

— isGoodAnswer : le booléen déterminant si la réponse du contributeur est juste

— certitude : certitude du contributeur en sa réponse

— time : le temps de réponse

— segmentDiff : la réponse à la question (q1)

— certaintyDiff : la réponse à la question (q2)

Considérons les graphiques des variables de ces ACP présentés figure 7. Le fait que le contributeur renseigne la bonne réponse (isGoodAnswer) à une question est fortement corrélé à la difficulté (valeur de delta) et à sa certitude que la réponse est juste. Ces variables sont en revanche totalement dé-corrélées de la perception que le contributeur a de sa capacité à renseigner le segment le plus long (segmentDiff) et de sa certitude (certaintyDiff) qui sont

corrélées entre elles. De plus la difficulté de la question est négativement corrélée au temps de réponse (time) du contributeur.

4 Modélisation des réponses

Dans cette section nous comparons 3 méthodes distinctes de modélisation et d'agrégation des réponses. La première utilise le vote majoritaire, la seconde une approche probabiliste et la troisième la théorie des fonctions de croyance.

Vote majoritaire Pour un vote majoritaire, les réponses des contributeurs sont modélisées par des fonctions indicatrices, telles que la réponse d'un contributeur c à la question q s'écrive r_{cq}. Soit Ω l'ensemble des réponses possibles à la question q, $r_{cq}(X) = 1, X \in \Omega$ si le contributeur choisit la réponse X, sinon $r_{cq}(X) = 0$. Les réponses sont agrégées en réalisant la somme des indicatrices, et la réponse ayant été donnée par le plus grand nombre de contributeurs est sélectionnée.

Probabilité On attribue à la réponse du contributeur la probabilité la plus haute, les autres éléments sont équiprobables avec $P_{cq}(Y) \leq P_{cq}(X)$.

$$\begin{cases} P_{cq}(X) = \alpha,\ X \in \Omega \\ P_{cq}(Y) = \frac{1-\alpha}{|\Omega|-1},\ \forall Y \in \Omega \setminus X \end{cases} \tag{1}$$

La valeur de la probabilité minimale α doit ici être supérieure ou égale à 0.2 pour que l'inégalité $P_{cq}(Y) \leq P_{cq}(X)$ soit vérifiée, car $|\Omega| = 5$. En revanche avec $\alpha = 0.2$ comme valeur numérique de l'incertitude totale du contributeur, nous obtenons une équiprobabilité totale représentant l'ignorance sur la réponse. Les valeurs numériques (α), associées à la certitude de la réponse du contributeur sont présentées dans le tableau 1.

Fonctions de croyance La théorie des fonctions de croyance a été introduite par Dempster (1967) et permet de modéliser l'incertitude et l'imprécision de sources d'informations imparfaites. Dans le cadre du *crowdsourcing*, les contributeurs sont des sources d'informations et leurs réponses dans nos expériences peuvent être imprécises et/ou incertaines. La théorie des fonctions de croyance considère un ensemble Ω d'hypothèses, celui-ci est appelé cadre de discernement. Des fonctions de masses $m : 2^{\Omega} \rightarrow [0, 1]$ sont alors définies telle que :

$$\sum_{X \in 2^{\Omega}} m(X) = 1 \tag{2}$$

L'ensemble des réponses proposées à un contributeur c pour une question q constitue le cadre de discernement Ω. Il est alors possible d'associer à la réponse X du contributeur une fonction de masse à support simple :

$$\begin{cases} m^{\Omega}_{cq}(X) = \alpha \text{ with } X \in 2^{\Omega} \setminus \Omega \\ m^{\Omega}_{cq}(\Omega) = 1 - \alpha \end{cases} \tag{3}$$

Totalement incertain	Incertain	Un peu incertain	Ni certain, ni incertain	Un peu certain	Certain	Totalement certain
0.2	0.3	0.4	0.5	0.6	0.7	0.8

TAB. 1 – *Valeur numériques α associées à l'échelle de certitude*

δ	0	300	600	900	1200	1500
MV	0	15	15	15	15	15
Probabilité	0	15	15	15	15	15
Fonctions de croyances	0	15	15	15	15	15

TAB. 2 – *Nombre de bonnes réponses (/15) pour Δ_0*

Cette fonction caractérise le fait que le contributeur croit partiellement en la réponse X avec une masse α mais pas davantage. L'ignorance est modélisée dans la théorie des fonctions de croyance par l'élément Ω. Notons que la réponse X du contributeur peut être imprécise. Les valeurs numériques de α utilisées sont les mêmes que pour notre modèle probabiliste (tableau 1).

Tests réalisés Une comparaison des trois méthodes énoncées ci-dessus est proposée pour l'agrégation des données de crowdsourcing. Pour ce faire, les réponses des contributeurs sont agrégées pour les 5 segments des 3 blocs pour chaque niveau de difficulté, soit un total de 15 valeurs agrégées par δ. Les tableaux 2 et 3 renseignent le nombre de fois où les réponses agrégées sont correctes sur les 15 questions considérées. Pour ces deux tableaux, il est constaté que les trois méthodes proposées offrent les mêmes résultats pour l'agrégation des réponses. Ceci montre que la majorité des contributeurs étaient bien attentifs et consciencieux dans la réalisation de la tâche.

5 Conclusion

L'ergonomie des tâches est une problématique au coeur des plateformes de crowdsourcing. De même, le recueil de données fiables et pertinentes est aussi recherché avec un plus grand pouvoir d'expression. Nous travaillons sur la définition d'une nouvelle interface ayant pour objectif de solutionner ces problèmes. Dans cette étude la première phase de notre protocole expérimental est abordée. Ainsi, nous proposons une interface permettant au contributeur de renseigner sa certitude en sa réponse.

Les tests de cette interface sont effectués sur des données réelles provenant du laboratoire pour certaines et d'une campagne de crowdsourcing pour d'autres. Une analyse détaillée des résultats montre que plus la tâche est simple plus le taux de bonnes réponses des contributeurs est important. De même, la certitude du contributeur augmente lorsque la tâche devient plus simple. A l'inverse, le temps de réponse des contributeurs décroît avec la difficulté de la tâche. De plus, d'après une analyse en composante principale, le taux de bonnes réponses, la difficulté de la tâche et la certitude renseignée par le contributeur sont fortement positivement corrélées,

δ	0	300	600	1200	1800	2400
MV	5	15	15	15	15	15
Probabilité	4	15	15	15	15	15
Fonctions de croyances	5	15	15	15	15	15

TAB. 3 – *Nombre de bonnes réponses (/15) pour* Δ_1

et négativement corrélées au temps de réponse du contributeur. Toujours d'après cette analyse, la perception de la difficulté de la tâche par le contributeur et sa capacité à renseigner sa certitude sont très fortement et positivement corrélés, et totalement indépendantes des autres variables énoncées précédemment. Il est également constaté d'après les résultats que le taux de bonnes réponses des contributeurs varie faiblement au début de l'expérience, puis se stabilise.

Nous avons modélisé les réponses des contributeurs de trois manières différentes : par vote majoritaire, par une approche probabiliste et par la théorie des fonctions de croyances. Il était supposé que la théorie des fonctions de croyance était l'approche la plus pertinente pour la modélisation des réponses mais ce n'est pas le cas ici puisque les trois méthodes comparées offrent les mêmes résultats. Ceci signifie que la majorité des contributeurs a raison, ceux-ci ont donc été attentifs et consciencieux dans la réalisation de la campagne de crowdsourcing.

Nos prochains travaux porteront sur la réalisation des deuxième et troisième phases expérimentales pour la définition de notre interface, portant respectivement sur l'imprécision, et l'incertitude combinée à l'imprécision.

Remerciements

Ce travail est en partie financé par le conseil départemental des Côtes d'Armor et a été réalisé dans le cadre d'un Projet Exploratoire Premier Soutien (PEPS).

Références

Abassi, L. et I. Boukhris (2018). A worker clustering-based approach of label aggregation under the belief function theory. *Applied Intelligence*, 1–10.

Ben Rjab, A., M. Kharoune, Z. Miklos, et M. Arnaud (2016). Characterization of experts in crowdsourcing platforms. *Belief Functions : Theory and Applications. 9861*.

Dempster, A. P. (1967). Upper and lower probabilities induced by a multivalued mapping. *The Annals of Mathematical Statistics 38*, 325–339.

Koulougli, D., A. Hadjali, et I. Rassoul (2016). Handling query answering in crowdsourcing systems : A belief function-based approach. In *Fuzzy Information Processing Society (NAFIPS), 2016 Annual Conference of the North American*, pp. 1–6. IEEE.

Ouni, H., A. Martin, L. Gros, M. Kharoune, et Z. Miklos (2017). Une mesure d'expertise pour le crowdsourcing. *Extraction et Gestion des connaissances (EGC)*.

Summary

Crowdsourcing is defined as the outsourcing of tasks to a crowd of contributors. The crowd is very diverse on these platforms and includes contributors with different skills and achieving the tasks more or less conscientiously. This profile diversity leads to uneven quality in the data. In order to overcome this problem we study a user interface offering the contributors a greater capacity of expression in their response. It alows the contributors to give imprecise answer in case of doubt and notified there confidence in their contributions. This interface was deployed asking users to answer at questions of variable complexity, for which ones these answers are known. Two range of complexity were tested and our results suggest that the user certainty depending on the absolute complexity of the question.

Atelier HIA - EGC 2020 :

Humain et IA : Travailler en intelligence

Présentation des plateformes Wirk.io &Yappers.club

Contexte :

Foulefactory est une plateforme de Crowdsourcing dédiée aux traitements dans des volumes importants de tâches à faible valeur ajoutée. Notre plateforme contributeur est fière aujourd'hui de revendiquer 50 000 contributeurs opérationnels et basés en France.

Grâce à nos travaux ANR + JEI des 2 années précédentes et notre expertise de plus de 5 ans dans le domaine du crowdsourcing, nous avons identifiés que :

- les pré-traitements par une IA (même peu performante) améliorent les performances des contributeurs humains (temps de traitements raccourcis, qualité finale de la donnée « revue »)

-La standardisation des instructions et l'ajout d'explications détaillées améliorent la qualité des traitements réalisés

Nous avons donc fait évoluer FouleFactory vers 2 nouvelles plateformes :

- Wirk.io : https://app.wirk.io, store permettant de lancer instinctivement des projets en intelligence artificielle ou collective
- Yappers.club : https://www.yappers.club, plateforme de travail collectif où opèrent nos 50 000 contributeurs

1 Présentation Wirk (https://app.wirk.io)

Wirk.io est un outil d'automatisation qui permet de simplifier plus de 400 cas d'usage sur différents types de traitement. Les tâches peuvent être effectuées par l'intelligence collectives via Yappers.club ou via l'utilisation de différents fournisseurs d'intelligence artificielle. Chaque projet est lancé via un template standardisé et simple d'utilisation que l'on appelle « App ».

1.0 Parcours du store

- Collection : Agrégation des meilleurs Apps par secteurs d'activités

- Recherche dans le store

- Présentation d'une App :

  - Cout par ligne de traitement, fonctionnement pas Saas

  - Type de traitement….

1.1 Lancement d'une App Auto

- Ajout des données

- Lancement d'une App IA traitement rapide : Catégorisation automatique d'images

1.2 Lancement d'une App Humaine

- Avec instruction

- et options paramétrables

- Mise en place des données

- Présentation de l'API

- Choix de la qualité : Différents choix de qualité par App, qui permettent de croiser les réponses d'un nombre variable de contributeurs, certifiés ou non.

2 Présentation Yappers.club (https://www.yappers.club)

Yappers.club, anciennement fouleurs.com est une plateforme de travail participatif regroupant 50 000 collaborateurs à travers toutes la France.

2.0 Parcours de la plateforme

- Projets : Affichage des projets auxquels le contributeur peut participer en fonction de son profil (segmentation, certifications, white liste et blackliste)

- Mes réalisations : historique de ses contributions

- Mes paiements & Mes informations : vérification KYC & payout

2.1 Certifications

Les certifications permettent d'évaluer les contributeurs sur un ensemble de compétences. Pour chaque certification, le contributeur doit répondre à un ensemble de questions/réponses. Ces certifications sont ensuite corrigées de manière manuelle ou automatique et donnent accès à des projets nécessitants l'obtention d'une note minimale à cette dernière.

Passage de certifications

2.2 Réponse à un projet :

Comparaison du template contributeur avec les données saisies sur Wirk

- Formulaire HTML/Formulaire standard
- Instructions saisies par sur Wirk
- Instructions standards complémentaires
- Options client
- Champs préformaté
- Temps de réalisation

2.3 Qualité et Validation croisée

Une fois la tache terminée, on opère un croisement des réponses entre contributeur (validation automatique : unanimité, majorité, totale)

Blacklist/WhiteList pour accéder à/exclure des projets

3 Perspectives : Wirk Desk

Fort de notre proposition de valeur, nous travaillons sur une nouvelle offre Wirk Desk, suite à la demande de clients d'utiliser le produit en « Marque Blanche », c'est-à-dire d'utiliser Wirk (les Apps

d’IA + l’API) mais en capitalisant sur les opérateurs de leurs back-offices internes : Le produit s’appelle Wirk Desk.

Cette nouvelle permettra d’optimiser la répartition du travail entre donneur d’ordre (manager) et opérateur, en rajoutant un nombre de métriques sur la performance et l’avancement des projets.

Summary

Proceedings of the Humans and AI (HIA) workshop, 2020.